Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow

Denis Fortun, Patrick Bouthemy, Charles Kervrann

Aggregation of local parametric candidates with exemplar-based occlusion handling for optical flow

Denis Fortun, Patrick Bouthemy*, Charles Kervrann*

Inria, Centre de Rennes - Bretagne Atlantique, Rennes, France

Abstract

Handling all together large displacements, motion details and occlusions remains an open issue for reliable computation of optical flow in a video sequence. We propose a two-step aggregation paradigm to address this problem. The idea is to supply local motion candidates at every pixel in a first step, and then to combine them to determine the global optical flow field in a second step. We exploit local parametric estimations combined with patch correspondences and we experimentally demonstrate that they are sufficient to produce highly accurate motion candidates. The aggregation step is designed as the discrete optimization of a global regularized energy. The occlusion map is estimated jointly with the flow field throughout the two steps. We propose a generic exemplar-based approach for occlusion filling with motion vectors. We achieve state-of-the-art results in the MPI-Sintel benchmark, with particularly significant improvements in the case of large displacements and occlusions.

Keywords: Optical flow, occlusion, large displacement, local parametric motion, aggregation framework.

1. Introduction

Optical flow is a key information when addressing important problems in computer vision such as moving object segmentation, object tracking, egomotion computation, obstacle detection or action recognition. The challenge for an optical flow estimation method is to deal with a large variety of image contents and motion types. Optical flow has been historically evaluated on sequences exhibiting small displacements and smooth motion fields, like in the Yosemite sequence [8]. Once initial issues were solved, other challenges were addressed [56], and new situations have been proposed by more recent benchmarks [5, 23]. Various and sometimes opposite scene conditions must be handled together, as illumination changes, large areas of smooth motion, motion details, large displacements, motion discontinuities, occluded regions (i.e., points disappearing in the next image).

Optical flow methods first rely on a data constancy assumption, e.g., applied to image intensity or spatial intensity gradient. Then, it is combined with a spatial, or sometimes space-time, coherency constraint on the expected velocity field. Existing approaches can be broadly classified into *local* and *global* methods.

Local spatial coherency arises when considering a parametric motion model, e.g., local translation [54], 4-parameter sub-affine model, affine model, 8-parameter quadratic model [61], in a given neighborhood or an appropriate local region. Optimization requires that the neighborhood is sufficiently textured or contains interest points such as corners, to supply accurate and reliable velocity vectors.

In contrast, global methods express the flow field coherency by imposing a global smoothness constraint in addition to the data constancy term, known as the regularization term of the global energy as pioneered by [39]. Global methods overcome uncertainty yielded by local supports in uniform intensity regions by diffusing motion from informative to non informative regions via the global regularization constraint. The optimization problem of seminal model [39] was optimally solvable, but the estimation was affected by oversmoothing and was limited to small displacements.

Numerous modifications of this original model, starting with [11, 38], have been designed to resolve these two crucial issues, namely, handling of large displacements and preservation of motion discontinuities. It was usually achieved by introducing a multi-resolution and incremental coarse-to-fine framework along with piecewise smoothing or robust estimation. A more recent attempt is to learn statistics of motion fields or motion bases as regularization means [44, 59, 67, 68, 84]. The data term of the global modeling has also received attention. Image features like image gradient [20], texture component [81], LDP (Local Directional Pattern) descriptor for illumination-robust data constancy [57], and matching criteria like Normalized Cross Correlation (NCC) [83] or Census transform [37], convey invariance properties to overcome limitations of the classical intensity constancy assumption. However, intricate optimization issues came with the increasing complexity of the modeling.

Although existing local methods are far from being able to compete with global models in terms of accuracy in computer vision benchmarks, several works based on joint estimation and segmentation of the motion field have shown that when appropriate segmented regions are found, affine models can be very accurate representations [73, 79]. Yet, the alternate optimiza-

*Corresponding author: Patrick.Bouthemy@inria.fr, tel. 33+299847274, fax 33+299847171

tion schemes involved are sensitive to the initialization of the region supports.

In this paper, we describe an occlusion-aware optical flow method that we name AggregFlow. It relies on an aggregation framework which explicitly separates the motion candidate computation and the global motion field recovery.

First, we advocate the systematic computation of affine motion models over a set of size-varying square patches, without segmentation step. To this end, we introduce a pre-defined collection of estimation windows to compute motion candidates, which allows us to seamlessly handle any configuration of piecewise continuous motions and a variety of motion scales. To handle large displacements, we combine affine estimation with patch-based matching. Differently from other methods exploiting feature matching as additional constraints [21, 82] or coarse initialization [26, 87], patch-based matching directly contributes to the computation of real-valued motion vector candidates at every pixel. We experimentally demonstrate that these sets of candidate motion vectors can potentially yield an accurate global flow field.

Secondly, we handle occlusion detection and occlusion filling with motion vectors in the two steps of AggregFlow. The set of motion candidates is in fact extended in two ways: exemplar-based search in occluded areas and use of the estimated parametric dominant motion. The local motion candidates are also exploited to build an occlusion confidence map which intervenes in the global aggregation model. We introduce a novel generic exemplar-based model for occlusion filling. It takes the form of an additional term in the global aggregation energy imposing non-local and image-based constraint on the motion of occluded pixels.

Thirdly, we resort to a discrete aggregation scheme. This kind of optimization approach has been little explored for optical flow computation so far [52], but it appears very promising. In coherence with the above observation about candidates accuracy, we define the aggregation as the selection of one motion candidate at each pixel, while ensuring smoothing of the resulting flow field and preservation of motion discontinuities. The aggregation is achieved with a discrete optimization algorithm, since motion candidates can be seen as labels. The occlusion confidence map enables to guide the joint occlusion and motion estimation, while decoupling the estimation of the two sets of unknown variables.

The main contributions of our method AggregFlow can be summarized as follows:

- An accurate parametric patch-based scheme for the motion candidate computation step with an efficient integration of feature matching,
- A generic exemplar-based approach for recovering motion in occluded regions,
- A joint motion field and occlusion map estimation guided by a local occlusion confidence map obtained from motion candidates.

We have carried out a comprehensive experimental evaluation. Specifically, state-of-the-art results have been obtained for large displacements and occlusions on the challenging MPI Sintel dataset. A preliminary approach without any occlusion handling and dedicated to a specific application was presented in [32].

The paper is organized as follows. Section 2 describes related work. In Section 3, we present the parametric computation of motion candidates and the local detection of occlusions. Section 4 is devoted to the aggregation stage. In Section 5, we report experimental results on three optic flow benchmarks, demonstrating the performance of AggregFlow. Section 6 contains concluding remarks.

2. Related work

Hereunder, we briefly review the literature on optical flow computation while focusing on issues related to our contributions. A recent comprehensive survey can be found in [33].

2.1. Feature correspondences and large displacements

The integration of feature correspondences in dense motion estimation has been investigated in several recent works. A first class of methods integrates feature correspondences in a global energy model. Variational methods [19, 21, 82] include an additional term to a classical global energy to force the flow to be close to pre-computed correspondences. Giving a fixed weight to the correspondences, this approach is sensitive to matching errors. To overcome this problem, [19, 66, 77, 82] focused on improving the matching step. Another class of methods use correspondences to reduce the search space for discrete optimization and provide a coarse initialization for subsequent refinement [26, 58, 87]. The main motivation of the attempts based on feature matching is to get rid of the drawbacks of the coarse-to-fine scheme imposed by variational optimization, in particular the loss of large displacements of small objects.

Our patch correspondence is related to [26, 58, 87] in the sense that it is used in the candidates generation process. However, our method does not produce coarse approximations to be refined in a continuous subsequent step and we do not perform any global variational optimization.

2.2. Occlusion handling

Occlusions play a crucial role for motion estimation [71], especially under large displacements, since no motion measurements are available in occluded areas. Therefore, a proper occlusion handling must distinguish between *occlusion detection*, segmenting the image into occluded and non-occluded regions, and *occlusion filling*, applying a specific treatment to motion estimation in occluded regions. Occlusion detection has been mostly undertaken as a subsequent operation to motion computation, by thresholding a consistency measure issued from the estimated motion field, like forward-backward motion mismatch [43], mapping unicity [87] or data constancy violation [85]. A distinctive geometric criterion is introduced in [46]. Occlusions can also be detected independently from motion estimation using image cues like spatiotemporal T-junctions [2]. In stereovision, other criteria like visibility [72] or ordering

constraints [17] are also exploited. The main limitation of this sequential approach is that accuracy of occlusion detection is highly dependent on the quality of the initial motion estimation. Several flow and image criteria have been combined in a learning framework [42]. Other approaches estimate the occlusion map jointly with the motion or disparity field in an alternate optimization scheme, encoding one of the above-mentioned criterion in a global energy [4, 43, 63, 72]. Our occlusion detection falls in the latter category.

The problem of filling occluded regions with estimated velocity vectors when the occlusion map is known is closely related to the image inpainting problem. Inpainting methods can be coarsely divided into two classes, diffusion-based methods [10, 24] and exemplar-based methods [28, 50]. A synthesis of these two approaches has been investigated in [22] in a variational framework. In exemplar-based image inpainting, the missing part is filled by copying pixels of the observed images. The framework is non local in the sense that similar pixels can be sought anywhere in the image. Occlusion filling is usually tackled by diffusion-based (or geometry-oriented) methods, propagating motion from non-occluded regions to occluded regions via partial derivative equation (PDE) resolution [4, 9, 43, 51, 63, 87]. In stereovision, a weighted least-squares strategy exploits a local averaging of the disparities of non-occluded pixels, with image- or segmentation-based weights [41]. It can be viewed as the local counterpart of the diffusion-based approach. In contrast, we adopt an exemplar-based strategy for occlusion filling with motion vectors, which could be called motion inpainting in occluded regions as well.

Another strategy is to handle occlusion detection and filling simultaneously with layered motion estimation [14, 74]. The depth information carried by the layered representation of motion encodes occlusions and disocclusions through relative displacements of overlapping layers.

2.3. Parametric motion estimation

The use of a parametric model has been widely investigated in motion estimation [12, 27, 32, 40, 55, 61, 73]. Applied on the whole image domain, affine or quadratic models are adequate to estimate the dominant image motion induced by the camera motion [61]. For accurate dense motion estimation, parametric approximations are only valid locally. Local regions are usually defined as square patches centered on each pixel [12, 54], possibly with an adaptation of the patch size [70], or its position [45]. It has the merit of being easy to implement with a low computational cost, but it is clearly outperformed by sophisticated extensions of [39] introduced in modern global optical flow methods.

More complex region shapes can be estimated by joint motion segmentation and estimation. Existing approaches can be divided in two classes. A first class of methods relies on an independent image color segmentation and tries to fit parametric motion in each region [13, 16, 36, 86, 91], possibly with the help of an independent global variational estimation [13, 86]. The drawback is that image color segmentation may lead to an over-segmentation of the motion field. The second class of methods jointly estimates supports of regions and parametric motion models for each region [27, 62, 73]. It is achieved by minimizing a global energy with respect to supports and motion parameters of the regions. However, the global energy is highly non-convex and particularly sensitive to the initialization of the optimization procedure.

The motion field produced by AggregFlow is composed of affine motion vectors estimated in square patches without any motion segmentation. AggregFlow implicitly selects the best patch size and position when selecting the best motion candidate for each pixel in the second step.

2.4. Discrete optimization and aggregation paradigm

Discrete optimization is an alternative to variational methods and is able to handle more general, non differentiable and non-convex, energy functionals. To combine the subpixel accuracy of the continuous variational approach and the efficiency of discrete minimization, the authors of [52] built a discrete motion space from motion fields delivered by several global variational estimations with different parameter settings. An energy function is then optimized by successive fusions of global proposals, which is efficiently achieved with a graph-cut technique. In [31], we followed a similar approach but with a semi-local patch-based variational estimation of candidate motion vectors. Recent works [26, 58] also exploit discrete graph-cut optimization in a two-step paradigm. However, the principle is different than ours. Indeed, the motion candidate generation step only aims to find dominant displacements and the aggregation provides a coarse initialization for a subsequent global refinement. In [34], belief propagation is used to minimize an energy with few candidates selected from a training set of image pairs chosen for their similarity with the input sequence. The dimensionality of the problem is further reduced by defining the graphical model over image patches rather than pixels. Discrete optimization is also associated with a variational framework in [87] as an intermediate stage between scales of a coarse-to-fine framework, in order to capture small objects lost in coarse scale levels. Aggregation in a variational framework has also been investigated in [1], where a set of candidate motion vectors is computed at each pixel using phase correlation in overlapping patches. The candidates are then linearly combined to create a global motion field. A similar approach has been explored for image colorization purposes in [64].

3. Local motion candidates and occlusion cues

We describe in this section the first step of our method AggregFlow. It exploits local information to supply motion candidates and occlusion cues. A set of motion vector candidates is generated at every pixel by a combination of patch correspondences and local parametric motion model estimates. A specific treatment is applied to occluded regions by exemplar-based extension of the motion candidates set. We also exploit the dominant motion in the image due to camera motion. Motion candidates and occlusion cues form the input of the second stage of AggregFlow described in Section 4.

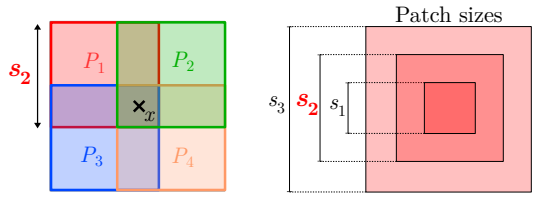


Figure 1: Four patches of set $\mathcal{P}_{s_2,\alpha}$ for a given size s_2 of the set $\mathcal{S} = \{s_1, s_2, s_3\}$, and overlapping ratio $\alpha = 0.3$. The pixel x is contained in the patches $P_1, \ldots, P_4$. Motion estimation in each of these patches provide motion candidates for x.

Our approach can be viewed as a new way to address the problem of choosing the local neighborhood for parametric estimation. Rather than adapting the regions *a priori* or jointly with the motion field, we operate in two steps: 1) estimation of motion candidates on several supports at every pixel, 2) implicit selection of the best support through the selection of the optimal candidate at each pixel within the aggregation step. In the sequel, we denote two consecutive image frames as $I_1, I_2 : \Omega \to \mathbb{R}$, with Ω denoting the image domain.

3.1. Local parametric motion candidates

3.1.1. Set of overlapping patches in I_1

The local supports for motion candidate computation are overlapping square patches of different sizes. Let us denote $\mathcal{P}_{s,\alpha}$ the patch set for a fixed patch size s and an overlapping ratio $\alpha \in [0, 1]$ indicating the proportion of surface shared by neighboring patches (see illustration of Fig.1). Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ be a set of n patch sizes, we then define $\mathcal{P}_{\mathcal{S},\alpha} = \bigcup_{s\in\mathcal{S}} \mathcal{P}_{s,\alpha}$. Due to the overlap and the number of patch sizes ($n > 1$), one given pixel $x \in \Omega$ belongs to several patches. The motion vectors are estimated independently in each patch in two sub-steps described below: patch correspondences and affine motion estimations.

3.1.2. Patch correspondences

For each patch $P_1 \in \mathcal{P}_{\mathcal{S},\alpha}$, we first determine the set $\mathcal{M}_N(P_1)$ of the N patches in I_2 most similar to P_1, which allows us to cope with arbitrarily large displacements. Let us put forward that we do not aim at keeping at this stage the best correspondence only but at selecting N relevant correspondences to subsequently constitute motion candidates. The matching step is generic and could be achieved with any arbitrary feature matching algorithm. We use a combination of the saturation and value channels of the HSV color space to gain partial robustness to illumination changes [90] and we use the Sum of Absolute Distances (SAD) to compare patches. The size of the reference patch and of the patches in the search area are the same. For each established pair of corresponding patches $P_{1,2} = (P_1, P_2)$ with $P_2 \in \mathcal{M}_N(P_1)$, we get the translation vector $\mathbf{w}_{P_{1,2}} \in \mathbb{Z}^2$ shifting P_1 onto P_2.

3.1.3. Affine motion refinement

The shift vectors obtained by the patch correspondence step capture large displacements, but they are not accurate enough to constitute a satisfying motion candidate set. First, they convey only integer-pixel accuracy, and secondly, they account for a local translation only inside each patch. To overcome these issues we refine the coarse displacement $\mathbf{w}_{P_{1,2}}$ by estimating a continuous, affine motion field $\delta\mathbf{w}_{P_{1,2}}$, independently for each pair of patches $P_{1,2}$. Denoting Ω_{P_1} the pixel sub-domain of P_1, the affine motion model $\delta\mathbf{w}_{P_{1,2}} : \Omega_{P_1} \to \mathbb{R}^2$ between patches P_1 and P_2 , which have the same size by construction, is defined at pixel $x = (x_1, x_2)^\top$ as:

$$\delta\mathbf{w}_{P_{1,2}}(x) = (a_1 + a_2x_1 + a_3x_2, a_4 + a_5x_1 + a_6x_2)^\top. \quad (1)$$

The parameter vector $\boldsymbol{\theta}_{P_{1,2}} = (a_1, a_2, a_3, a_4, a_5, a_6)^\top$ of the affine model is estimated assuming brightness constancy and applying first the coarse registration given by $\mathbf{w}_{P_{1,2}}$:

$$\widehat{\boldsymbol{\theta}}_{P_{1,2}} = \arg\min_{\boldsymbol{\theta}_{P_{1,2}}} \int_{\Omega_{P_1}} \psi(P_2(x + \mathbf{w}_{P_{1,2}} + \delta\mathbf{w}_{P_{1,2}}(x)) - P_1(x))dx \quad (2)$$

where the penalty function $\psi(\cdot)$ is chosen as the robust Tukey's function. The problem (2) is solved with the publicly available Motion2D software[1] [61], which implements a multi-resolution incremental minimization scheme involving an IRLS (Iteratively Reweighted Least Squares) technique. The algorithm is initialized by setting affine parameters to zero, and the non-convexity of the objective function in (2) is handled with a graduated non-convexity approach [15] iteratively adapting the parameter of the Tukey function.

3.1.4. Final set of motion candidates

The above described two-step estimation is repeated for every patch of $\mathcal{P}_{\mathcal{S},\alpha}$ and generates a set of candidate motion vectors $C(x)$ at each pixel $x \in \Omega$ defined as follows:

$$C(x) = \{\mathbf{w}_{P_{1,2}}(x) + \delta\mathbf{w}_{P_{1,2}}(x) \quad (3)$$
$$: P_1 \in \mathcal{P}_{\mathcal{S},\alpha}(x), P_2 \in \mathcal{M}_N(P_1)\},$$

where $\mathcal{P}_{\mathcal{S},\alpha}(x) = \{P \in \mathcal{P}_{\mathcal{S},\alpha} : x \in P\}$.

Let us make a few comments on the estimation scheme for computing motion candidates. A coarse motion estimation followed by a refinement step has been investigated in several previous works [26, 53, 58], but it has always been dedicated to global motion fields. In our case, the refinement is local and adapted to each patch correspondence. Classical local motion estimation methods based on [54] also rely on square patches, but assign the computed motion vector only to the center point of each patch. On the opposite, parametric motion estimation in segmented regions as in [27] applies to regions of arbitrary shape. Our patch distribution can be considered as an intermediate level between these two extremes. Indeed, we use square patches as in [54] and thus avoid the complex segmentation

[1] http://www.irisa.fr/vista/Motion2D/

step. However, we exploit the whole vector subfield issued from the affine model estimated in each patch. As a consequence, every pixel inherits several motion candidates from the affine motion estimations performed in patches of different positions and sizes containing this pixel. Finally, in contrast to several other methods using feature correspondences [21, 26, 82], we do not select one single patch correspondence but we keep the N best ones.

The advantages of the local sets of motion candidates supplied by AggregFlow are three-fold. First, the correspondence sub-step enables us to capture large displacements even for small patch sizes. Thus, it allows us to correctly deal with small structures undergoing large displacements in contrast to coarse-to-fine schemes. Second, by considering a large variety of patches, we get rid of the predefined choice of the local neighborhood encountered in parametric motion estimation. The selection of the proper patch via its corresponding motion candidate is transferred to the aggregation stage. Third, introducing patches of several sizes enables us to tackle motion of different scales.

3.2. Motion candidates in occluded areas

The generation of motion candidates described in Section 3.1 does not differentiate between occluded and non-occluded pixels. For a given pixel x, if all the patches of $\mathcal{P}_{S,\alpha}(x)$ mainly contain occluded pixels, there is no chance to correctly estimate a relevant motion candidate at x. Therefore, we compute additional motion candidates in occluded regions in a specific manner.

Let us define the occlusion map $o : \Omega \to \{0, 1\}$

$$o(x) = \begin{cases} 1 & \text{if } x \text{ is occluded,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The occluded regions are denoted $O = \{x \in \Omega : o(x) = 1\}$. The computation of map o will be addressed in Section 3.4 and Section 4, and we assume for now that o is known.

3.2.1. Occlusion filling with motion vectors

When occluded regions are known, occlusion filling with motion vectors is conceptually closely related to image inpainting, since it recovers motion in regions where motion is by definition *not observable*. The occluded pixels do not appear in the next image and consequently have no corresponding points. Classical methods for motion-based occlusion filling operate in a variational framework by cancelling the data term and letting the diffusion process of the regularization propagate the optical flow in occluded regions [4, 87]. The diffusion-based class of inpainting methods [10] acts similarly. They perform well in case of thin missing areas or cartoon-like images, but they are usually outperformed by exemplar-based inpainting methods [28] for large missing regions. In order to deal with large occlusions produced by large displacements, we follow the inpainting analogy and we overcome the problem of local motion candidates estimation in occluded areas by designing an exemplar-based scheme to recover relevant motion candidates. In the first step of AggregFlow, the motion candidates set is thus augmented by copy-paste operations.

3.2.2. Exemplar-based candidates extension

We rely on the assumption that motion at an occluded pixel $x \in O$ is similar to the motion of a close non-occluded pixel $m_o(x) \in \Omega \backslash O$ belonging to the same object or the same background part. To provide relevant motion candidates at x, we copy motion candidates from $C(m_o(x))$ to $C(x)$. $m_o(x)$ is sought in a domain $\mathcal{V}_o \subset \Omega \backslash O$ which is close to the occlusion boundaries. Figure 2(e) represents the occluded regions O (in white) and the search domain $\mathcal{V}_o$ (in red), and Fig.2(f) superimposes the two sets on I_1. Searching for pixel $m_o(x)$ for $x \in O$ is actually easier for occlusion filling than for image inpainting. Indeed, occluded regions are not completely uninformative, while inpainted regions are, since we have access to the information supplied by image I_1 even in O. Thus, as $m_o(x)$ is expected to belong to the same object as x, we use color similarity to find the match in I_1:

$$m_o(x) = \underset{y \in \mathcal{V}_o}{\arg\min}\, D(I_1, x, y), \tag{5}$$

where $D(I_1, x, y)$ is the distance between patches centered respectively in x and y. As in Section 3.1, we resort to a SAD in the HSV space.

An extended candidate set $C_+(x)$ is created for occluded pixels by adding to the initial set $C(x)$ the motion candidates of their matched pixel $m_o(x)$:

$$C_+(x) = C(x) \cup C(m_o(x)),\ \forall x \in O. \tag{6}$$

A more sophisticated addition process could even be envisaged. We could take the velocity vectors provided at x by the parametric models corresponding to the motion candidates of $m_o(x)$. By convention, $\forall x \in \Omega \backslash O$, $C_+(x) = C(x)$.

3.2.3. Occlusions due to camera motion

A particular class of occluded (or disappearing) regions occurs at image borders in the case of large camera motion (Fig.3). We cope with this issue by estimating the dominant image motion due to camera motion. To do so, we use again the robust parametric estimation described in Section 3.1, but now, we apply it to the whole image [61], to retrieve the dominant motion. We found in our experiments that the quadratic model was more adequate to accurately cope with large and sometimes complex camera motion. The velocity vector supplied at x by the estimated parametric model of the dominant motion, $\mathbf{w}_{cam} : \Omega \to \mathbb{R}^2$, is added to the motion candidates of x.

We end up with the final overall set C_f of motion candidates:

$$C_f = \{C_f(x),\ x \in \Omega\}, \tag{7}$$

with $C_f(x) = C_+(x) \cup \{\mathbf{w}_{cam}(x)\}$. The camera motion candidates are mostly useful for occluded pixels, but it can sometimes provide relevant motion candidates in unoccluded regions of the background as well, so that we finally add it to all pixels in Ω.

Figure 2: Illustration of the performance improvement with exemplar-based candidates extension (without the dominant motion extension). First row: two successive input images. Second row: ground-truth occlusion map and motion field. Third row: representation of the search domain $\mathcal{V}_o$ (displayed here after median filtering of the occlusion map for the sake of visibility only). Fourth row: Best Candidate Flow obtained respectively without and with the exemplar-based candidates extension.

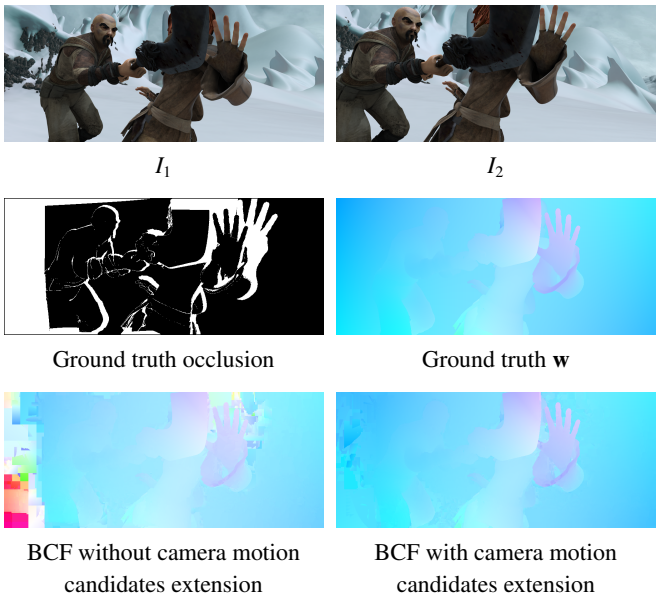


Figure 3: Performance improvement with camera motion candidates extension (without the exemplar-based candidate extension in occluded regions). First row: two successive input images. Second row: ground-truth occlusion map and motion field. Third row: Best Candidate Flow obtained respectively without and with the camera motion candidates extension.

Table 1: EPE-all scores of motion fields on sequences with ground-truth from MPI Sintel and Middlebury datasets

	MPI Sintel	Middlebury
Full BCF	**0.792**	**0.071**
BCF w/o candidates extensions	1.851	0.083
EpicFlow [66]	2.641	0.308
DeepFlow [82]	4.691	0.386
MDP-Flow2 [87]	4.006	0.223

3.3. Best candidate flow

To validate our method for computing motion candidates, we have processed sequences from MPI Sintel and Middlebury datasets [5, 23] provided with ground truth. We create the *Best Candidate Flow* (BCF) by selecting at each pixel x the candidate motion vector of $C_f(x)$ closest to the ground-truth vector. In order to evaluate our occlusion module, we distinguish between the BCF determined with the candidates extension described in the preceding section (or full BCF) and the BCF without it. Parameters involved in the local motion computation are set to $\mathcal{S} = \{16, 44, 104\}$, $\alpha = 0.75$, $N = 2$.

Illustrations of the contribution of motion candidate extensions are provided in Fig.2 and Fig.3 on sequences of the MPI Sintel benchmark involving large occluded regions. The difference between BCF without any candidate extension and the full BCF is clearly visible for occluded pixels and testifies the importance of the exemplar-based and camera motion candidate extensions. Overall, the full BCF is very close to the ground-truth motion field revealing the performance of the local parametric motion computation in the first step of AggregFlow.

We report in Table 1 the objective evaluation given by the Endpoint Error (EPE) scores for the full BCF and BCF without candidate extensions, on the sequences provided with ground-truth in the datasets MPI Sintel and Middlebury. We also compare them with those of motion fields supplied by [66, 82, 87], as obtained with publicly available code. Both BCFs outperform state-of-the-art methods [66, 82, 87] on the Sintel sequences with ground truth, and also performs better than these four methods on the Middlebury examples provided with ground truth. Accuracy is further significantly improved with full BCF, especially for the MPI Sintel sequences where large displacements and wide occluded regions are present. It demonstrates that the combination of local affine estimations in square patches with patch correspondences as described in Section 3.1, is quite relevant and sufficient to recover very accurate motion fields. The challenge now is to select the best velocity vector among the motion candidates at every pixel.

3.4. Occlusion confidence map

In Section 3.2, the occlusion map o was assumed to be known, and we addressed the motion-based occlusion filling problem by recovering motion candidates for occluded pixels from non-occluded areas. The occlusion detection task, that is the determination of o, will be performed through the two steps of AggregFlow. In the first step, we compute a coarse occlusion

confidence map, which will be used in the aggregation to guide the estimation. Our procedure is simple and exploits the patch distribution $\mathcal{P}_{\mathcal{S},\alpha}$ and the correspondences used for motion candidates estimation. Nevertheless, from a more general point of view, the coarse occlusion confidence map could be designed differently, e.g., in the framework of [47].

We first perform a coarse occlusion detection at the patch level. We consider the smallest patch size s_1 of the set $\mathcal{S}$ defined in Section 3.1 and detect the occluded patches of the set $\mathcal{P}_{s_1,\alpha}$. A common and simple occlusion detection consists in checking the consistency of forward and backward estimated motion vectors [42, 43, 58]. We apply the same principle to patches of $\mathcal{P}_{s_1,\alpha}$. Simplifying the notations of Section 3.1 for the sake of readability, let us denote T_P^f the forward translation between a patch $P \subset I_1$ and its matched patch $M_P \subset I_2$, and T_P^b the backward translation between M_P and its matched patch in I_1. The forward-backward consistency criterion states that the patch P is occluded if $\|T_P^f + T_P^b\| > \nu$, where ν is a threshold. We then infer a patch-based occlusion map $o_\wp$ as follows:

$$o_\wp(x) = \begin{cases} 1 & \text{if } \exists P \in \mathcal{P}_{s_1,\alpha}(x) \text{ such that } P \text{ is occluded} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Let us now consider the point set $\mathcal{X}_{o_\wp}$ composed of the centers of the occluded patches. We use the density of the point set as an indicator of the presence of occlusions. We apply a Parzen density estimation on $\mathcal{X}_{o_\wp} = \{x_1, \ldots, x_{N_\wp}\}$, with $N_\wp$ the number of occluded patches:

$$\omega_o(x) = \frac{1}{N_\wp} \sum_{i=1}^{N_\wp} \frac{1}{s_1} K\left(\frac{x - x_i}{s_1}\right), \tag{9}$$

where K is a Gaussian kernel. We take the bandwith equal to s_1 to be coherent with the first step of motion candidate computation. The occlusion confidence map ω_o is thus built as a probability density of the occlusion state. The closer to 1 the value of $\omega_o(x)$ the more likely the presence of an occluded point at x. Figure 4 shows an example of $o_\wp$ and ω_o. The preliminary occlusion map $o_\wp$ is precisely the map used in the first step of AggregFlow for the exemplar-based candidates extension in occluded regions as described in Section 3.2.2. The map ω_o is exploited in the aggregation stage to guide the sparsity-constrained occlusion reconstruction.

The output of the first step of AggregFlow are the overall set C_f of motion candidates and the occlusion confidence map ω_o. Then, they are used as input of the second step of AggregFlow, that is, the aggregation, to recover the global motion and occlusion fields. It is described in the next section.

4. Discrete aggregation

The set of motion candidates at a given pixel is formed by a finite (discrete) set of vectors, but the motion vectors themselves are computed in the continuous space $\mathbb{R}^2$ with the affine motion refinement. The final motion vectors are selected among the motion candidates. Since the motion candidates set comprises a finite number of vectors, it can be seen as a discrete set of labels allowing for discrete optimization. The analysis of the Best Candidate Flow in subsection 3.3 has shown that the set of candidates at each pixel generally contains at least one motion vector very close to the ground truth. Therefore, we view the aggregation as the selection of the best candidate at every pixel.

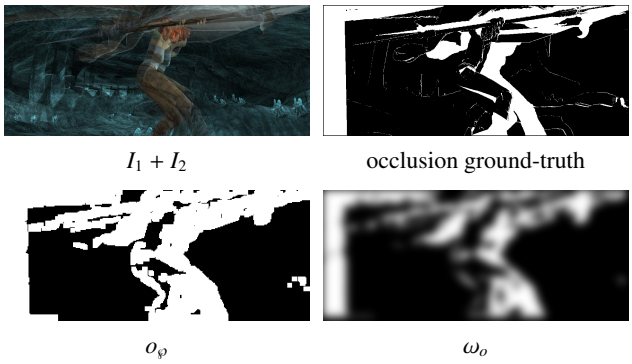


Figure 4: Patch-based occlusion detection. First row: Overlay of the two successive input images and occlusion ground-truth. Second row: Corresponding computed patch-based occlusion map $o_\wp$ and occlusion confidence map ω_o.

To this end, we formulate the aggregation as a discrete optimization problem, where the discrete finite motion vector space at each pixel x is composed of the motion candidates $C_f(x)$. The occlusion map will be estimated jointly with the motion field while exploiting the occlusion confidence map ω_o. The aggregation step amounts to minimizing the global energy function $E(\mathbf{w}, o)$:

$$\begin{aligned} \{\widehat{\mathbf{w}}, \widehat{o}\} &= \underset{\{\mathbf{w},o\}}{\arg\min}\ E(\mathbf{w}, o) \\ &\text{s.t.} \quad \mathbf{w}(x) \in C_f(x),\ o(x) \in \{0, 1\}. \end{aligned} \tag{10}$$

In the following, we detail the design of $E(\mathbf{w}, o)$ and the optimization strategy we have adopted.

4.1. Global energy definition

The aggregation energy is composed of four terms:

$$\begin{aligned} E(\mathbf{w}, o) &= E_{data}(\mathbf{w}, o, I_1, I_2) + E_{occ}(o, \omega_o) \\ &+ E_{reg_w}(\mathbf{w}) + E_{reg_o}(o). \end{aligned} \tag{11}$$

We now describe in turn each term of the energy function $E(\mathbf{w}, o)$.

4.1.1. Data term E_{data}

The data term accounts for the relations between motion, occlusion and input images. At non-occluded pixels, i.e., $o(x) = 0$, we rely on the usual constancy assumption of image intensity and of spatial image gradient, and we robustly penalize the deviation from the data constraints. The potential ρ_{vis} associated to non-occluded (or visible) pixels is given by:

$$\begin{aligned} \rho_{vis}(x, \mathbf{w}) &= \phi(I_2(x + \mathbf{w}(x)) - I_1(x)) \\ &+ \gamma\ (\phi(\nabla_{x_1} I_2(x + \mathbf{w}(x)) - \nabla_{x_1} I_1(x)) \\ &+ \phi(\nabla_{x_2} I_2(x + \mathbf{w}(x)) - \nabla_{x_2} I_1(x))), \end{aligned} \tag{12}$$

where ϕ is the L_1 norm, γ balances intensity and gradient constancy potentials, and $\nabla_{x_k} I_i$ denotes the partial derivative of each

image $I_i, i = 1, 2$, w.r.t. each image coordinate $x_k, k = 1, 2$. Resorting to discrete optimization allows us to use the non-linearized brightness constancy equation. Thus, coarse-to-fine scheme is not required to cope with large displacements, and we avoid drawbacks related to the loss of small objects with large displacements.

At occluded pixels, no correspondence can be established by definition, and consequently no image feature constancy constraint can be exploited. Therefore, consistently with the motion candidate extension of the first step, we define an exemplar-based data term for occluded pixels, encoded in the potential ρ_{occ}:

$$\rho_{occ}(x, \mathbf{w}, m) \;=\; \|\mathbf{w}(x) - \mathbf{w}(m(x))\|^2 \,, \tag{13}$$

where $m(x)$ is the visible pixel matched with pixel x as obtained in (5). The selected motion vector at an occluded pixel is thus expected to be similar to the selected motion vector of its matched non-occluded pixel. The data term is finally formed by incorporating the selection of either the visible or the occlusion potential using the occlusion map:

$$\begin{aligned} E_{data}(\mathbf{w}, o, I_1, I_2) = \sum_{x\in\Omega} & (1 - o(x))\, \rho_{vis}(x, \mathbf{w}) \\ & +\; \lambda_1\, o(x)\, \rho_{occ}(x, \mathbf{w}, m). \end{aligned} \tag{14}$$

In contrast to other occlusion handling schemes in optical flow methods which only cancel the visibility term ρ_{vis} in occluded areas and fill the occlusions with motion vectors by diffusion [4, 63, 87], ρ_{occ} acts as a valid exemplar-based data term at occluded pixels.

Concerning the occlusion recovery (i.e., the optimization w.r.t. o), the data term favors the selection of the occluded label at pixels where the data constancy term is strongly violated. The continuous approach of [4] operates in a similar way. In [4], the data constancy deviation is balanced by an estimated continuous residual intensity field, from which occluded points are retrieved by thresholding. In contrast, our occlusion map is binary by nature, and strongly prevents the influence of irrelevant data-constancy constraints on motion estimation in occluded areas.

4.1.2. Occlusion term E_{occ}

The data term (14) favours the detection of occluded pixels and must be counterbalanced by another term penalizing occlusion occurrence defined by:

$$E_{occ}(o, \omega_o) = \lambda_2 \sum_x (1 - \omega_o(x)) o(x), \tag{15}$$

where ω_o is the occlusion confidence map computed in the first stage. The penalty of occlusion occurrence can be interpreted as a sparsity constraint on the binary occlusion field o. A sparsity constraint for occlusion detection was also proposed in [4] in a continuous setting, and in [63] for a binary occlusion variable, but without confidence map.

If we set $\forall x \in \Omega, \omega_o(x) = 0$, which would be similar to what is done in [4, 63], the data-driven occlusion detection would boil down to the data term (14), while (15) would be a pure sparse prior constraint. The detection of the occlusion map would be then too tightly coupled with the currently estimated motion field. We would face a chicken-and-egg problem, where o is determined by $\mathbf{w}$, which also depends on o. The consequence of the alternate optimization scheme would be a rapid trap into a local minimum. This issue and the benefit yielded by our weighting strategy are illustrated in Section 5.

4.1.3. Regularization terms E_{reg_w} and E_{reg_o}

The term $E_{reg_w}(\mathbf{w})$ enforces piecewise smoothness of the motion field:

$$E_{reg_w}(\mathbf{w}) = \lambda_3 \sum_{<x,y>} \beta(x) \phi(\mathbf{w}(x) - \mathbf{w}(y)) \tag{16}$$

where ϕ is the L_1 norm, $< x, y >$ denotes the two-site clique issued from the 8-neighborhood system. The weights $\beta(x)$ are specified as $\beta(x) = \exp\left(-\|\nabla I_1(x)\|^2/\tau^2\right)$ to modulate the regularization according to the intensity edge strength.

It is also important to impose smoothness of the occlusion map with the term E^o_{reg}:

$$E_{reg_o}(o) = \lambda_4 \sum_{<x,y>} (1 - \delta(o(x) = o(y))), \tag{17}$$

where δ designates the Kronecker function equal to 1 if its argument is true.

4.2. Optimization

The optimization problem (10) is solved by alternating minimization w.r.t. $\mathbf{w}$ and o. The initial value of o is given by the coarse patch-based occlusion detection o_φ defined in (8). The set m of matching points attached to the exemplar-based candidates extension is initialized with m_o defined in (5). It is recomputed after each update of the occlusion map. Table 5 gives an overview of AggregFlow method.

4.2.1. Optimization specifications

Hereafter, we give details on the minimization procedure concerning $\mathbf{w}$ and o. Once $\widehat{\mathbf{w}}$ is fixed, the energy to optimize w.r.t. o amounts to:

$$\begin{aligned} \min_o \sum_{x\in\Omega} & (1 - o(x))\, \rho_{vis}(x, \widehat{\mathbf{w}}) \;+\; \lambda_1\, o(x)\, \rho_{occ}(x, \widehat{\mathbf{w}}, m) \\ & +\; \lambda_2 \sum_x \omega_o(x) o(x) \;+\; \lambda_4 \sum_{<x,y>} (1 - \delta(o(x) = o(y))). \end{aligned} \tag{18}$$

Since the pairwise term is submodular, the problem (18) can be solved exactly with standard graph cut method [18].

The optimization w.r.t. $\mathbf{w}$ with $\widehat{o}$ fixed is more difficult. The reduced energy function writes:

$$\begin{aligned} \min_{\mathbf{w}} \sum_{x\in\Omega} & (1 - \widehat{o}(x))\, \rho_{vis}(x, \mathbf{w}) \;+\; \lambda_1\, \widehat{o}(x)\, \rho_{occ}(x, \mathbf{w}, m) \\ & +\; \lambda_3 \sum_{<x,y>} \beta(x) \phi(\|\mathbf{w}(x) - \mathbf{w}(y)\|^2). \end{aligned} \tag{19}$$

The global motion label space C_f has the specificity to be huge and space-variant. Indeed, the size of each individual set $C_f(x)$

can already exceed 200, and by construction the content of $C_f(x)$ depends on x. Message passing methods like belief propagation [30] and TRW-S [49] can be applied to spatially varying label sets, as investigated in [80] for stereo, but we found these methods to be too slow for our minimization problem (19). An alternative is to resort to graph-cut move-making methods [18], generalized in [52] to spatially varying label sets. In this setting, each *move* is a binary optimization problem defined on an auxiliary variable selecting between two global proposals. Due to the spatial variability of the proposals and their independence, the submodularity of the regularization potential of (19) cannot be ensured, and only suboptimal moves can be achieved using QPBO [69].

Another issue arises from the non-local interaction involved in the exemplar-based term $\rho_{occ}(x, \mathbf{w}, m)$. To make the optimization problem tractable, we transform $\rho_{occ}(x, \mathbf{w}, m)$ to a pixel-wise term at each move-making iteration by fixing the exemplar-based constraint $\mathbf{w}(m(x))$ to its value at the previous iteration. At a given move-making iteration i, denoting $\widehat{\mathbf{w}}^{(i-1)}$ the value of $\mathbf{w}$ at iteration $i-1$, the potential becomes:

$$\rho_{occ}(x, \mathbf{w}, m) = \left\| \mathbf{w}(x) - \widehat{\mathbf{w}}^{(i-1)}(m(x)) \right\|^2 . \qquad (20)$$

4.2.2. Proposal construction

Our aggregation problem differs from the one of [52] since our motion candidates are locally determined. In contrast, [52] exploits global flow fields that can be directly used as proposals in the move-making process. Thus, we have to build global flow field proposals at each iteration from the local motion candidates computed in patches. The important point is to ensure spatial smoothness of the proposals, in accordance with the regularization term of the model (19). Therefore, we build a global flow field proposal by considering a tiling of non-overlapping patches of a given size and by selecting at every pixel in each patch the motion candidate precisely issued from that patch. This construction maintains the spatial coherency of the local affine estimations. We build as many global proposals as necessary to reasonably explore the motion candidate space.

5. Experimental results

In this section, we assess the performance of AggregFlow with experiments on several optic flow benchmarks and we deeply analyse the contribution of AggregFlow in occlusion areas.

5.1. Implementation and parametrization

First, we provide information on implementation and parametrization issues.

All the patch correspondences involved in AggregFlow are computed with the PatchMatch algorithm [7] based on the minimal C++ code provided by the authors[2]. For the discrete minimization, we use available QPBO and max-flow code[3].

[2] http://gfx.cs.princeton.edu/pubs/Barnes_2009_PAR/index.php

[3] http://pub.ist.ac.at/ vnk/software.html

1. Local step

- **1.1.** Generate the motion candidates sets $C(x)$ (3)
- **1.2.** Compute patch-based occlusion map $o_{\wp}$ (8)
  Derive the occlusion confidence map ω_o (9)
- **1.3.** Compute the matching variables $m_o(x)$ (5)
  Extend motion candidates to obtain C_f (7)

Output of the 1^{st} step: C_f, ω_o

2. Global aggregation
Initialize $o = o_{\wp}$ and $m = m_o$
Iterate:

- **2.1.** Estimate $\mathbf{w}$ (19)
- **2.2.** Estimate o (18)
- **2.3.** Update m (5)

Output of the 2^{nd} step: $\mathbf{w}$, o

3. Post-processing : weighted median filtering on $\mathbf{w}$

Table 2: Overview of AggregFlow

The three datasets used for the experimental evaluation exhibit very different motion characteristics. The Middlebury dataset involves small displacements, and motion discontinuities often occur. The KITTI dataset only contains sequences acquired with a camera embedded in a moving car, which produces smooth and diverging motion fields. Finally, the MPI Sintel benchmark is the most challenging one, with very large displacements and occlusions, and a large variety of motion types, from complex smooth deformation to discontinuous piecewise constant motion. Therefore the parameters of the method have to be adapted to obtain optimal results. Parameter λ_1 does not have a decisive influence since it aims to give comparable range to the data term at occluded and non-occluded pixels. Parameter λ_2 controls the amount of detected occluded pixels. In presence of large occluded regions, data conservation is often violated and λ_2 should be set large enough to counterbalance this effect. Parameter λ_3 is a classical regularization parameter on the motion field. Parameter λ_4 accounts for the strength of the regularization of the occlusion map, it should be high enough when occlusion regions are large. After extensive experimental tests, the aggregation parameters have been set to $\lambda_1 = 5$, $\lambda_2 = 50$, $\lambda_3 = 500$, $\lambda_4 = 20$ for for all the image sequences of the MPI Sintel benchmark, to $\lambda_1 = 2$, $\lambda_2 = 10$, $\lambda_3 = 250$, $\lambda_4 = 4.5$ for all the image sequences of the Middlebury dataset and to $\lambda_1 = 2$, $\lambda_2 = 10$, $\lambda_3 = 500$, $\lambda_4 = 30$ for the KITTI dataset. The determination of the corresponding visible points $m(x)$ is performed with patches of size 11×11.

To capture different motion scales, the patch sizes must cover a sufficient range of values. In all our experiments, we will use

$\mathcal{S} = \{16, 44, 104\}$. To avoid that the set $\mathcal{M}_N(P_1)$ uselessly contains too close patches, we impose a minimal distance between two patches of $\mathcal{M}_N(P_1)$.

For the initial patch-based detection of occluded points, the threshold ν is set to 10. In the exemplar-based motion candidates extension, the domain $\mathcal{V}_o$ is obtained by dilating the occluded regions by 20 pixels and by taking the band given by the dilated regions minus the original ones.

We fix the number of overall updates on the motion field $\mathbf{w}$ and the occlusion map o to 3 for all sequences to save computation time. Indeed, convergence was empirically observed after three iterations in most cases. A weighted median filtering with bilateral weights [88] is performed on the computed motion field as a post-processing step to enhance motion edges as advocated in [75].

As a representative example, the computation time for the *Urban2* sequence of the Middlebury benchmark (640×480 pixels) is 27 minutes on a Intel Xeon laptop with 2.20GHz clock speed and 64Gb RAM. More precisely, the first step of candidates computation takes 10 minutes, the global optimization step 15 minutes, and the weighted median filter 2 minutes. Most of the computation time is consumed in the patch correspondence sub-step for the largest patch size (106×106 pixels). We have not dedicated specific effort to optimize the code so far. Computation time is higher than for most variational approaches, but it could be reduced by some simple implementation tricks. For instance, the first step of AggregFlow can be massively parallelized on GPU. The correspondence step could be handled in a different way for the largest patch size, by downsampling the patches for instance. Besides, the use of integral images for matching can significantly accelerate computation [29]. Alternative algorithms for parametric image registration could also be envisaged [76]. Fast weighted median filter [92] can also be exploited in the post-processing step. Another strategy to speed-up the aggregation step could be to reduce the size of the set C_f by adding a pruning step based on a simple criterion like forward/backward consistency [43].

5.2. Quantitative evaluation on optic flow benchmarks

We have evaluated AggregFlow on three optical flow benchmarks: MPI Sintel flow dataset[4] [23], Middlebury flow dataset[5] [5], and KITTI dataset[6] [35]. The MPI Sintel benchmark is the most relevant one to assess AggregFlow performance since it involves wide occlusion areas and large displacements, which are precisely the issues on which AggregFlow is claimed to bring significant contributions. The Middlebury benchmark offers other challenges as preservation of motion details. The KITTI dataset is not as generic as the two first datasets for evaluating optic flow methods. Indeed, it delivers very specific diverging motion fields since the camera is mounted on a moving vehicle and observes static street scenes. We have retained the Endpoint Error measure (EPE) for quantitative evaluation.

MPI Sintel flow dataset Sequences of the MPI Sintel benchmark [23] are characterized by long-range motion, motion blur, non-rigid motion, and wide occluded areas. Methods are evaluated on two versions of the sequences named *Clean* and *Final*. The Final version adds motion and defocus blur along with atmospheric effects like fog on some sequences. We reproduce in Table 3 the top 12 published methods (including ours at paper submission date) in the MPI Sintel benchmark for the Clean set. Table 4 contains the performance of the same twelve methods for the Final set. Results are analyzed through several indicators: "EPE all" is the average EPE on all the sequences; "EPE matched" and "EPE unmatched" restrict the error measure respectively to regions that remain visible in adjacent frames (non-occluded pixels) and to regions that are visible only in one of two adjacent frames (occluded pixels); "d0-10" denotes EPE over regions closer than 10 pixels to the nearest occlusion boundary, and thus reveals the ability to recover motion discontinuities; "s40+" denotes EPE over regions with velocities larger than 40 pixels per frame. Methods are ranked regarding their EPE all.

We first conducted experiments on MPI Sintel sequences provided with ground truth. Results on four sample sequences are displayed both for motion field estimation and occlusion map determination in Fig.5. Visual comparison with motion fields estimated with the state-of-the-art methods [66] and [87] is also provided. They have been obtained with the public codes provided by the authors[7,8]. These results will be commented hereunder within the discussion on Table 3 and 4.

For the *Clean* set, our method AggregFlow ranks third over the published methods. The competitive performance on the unmatched category (ranked third) emphasizes the efficiency of our occlusion framework. AggregFlow is ranked sixth for the d0-10 metric (but very close to PH-Flow [89] ranked third), which demonstrates its capacity to recover motion discontinuities as confirmed by results displayed in Fig.5. First, it is due to the robust affine estimation of the motion candidates able to capture locally dominant motion in case of two or more motions present inside patches. It is also made successful by the efficient occlusion module, which allows us to moderate the need for motion field regularization. Indeed, missing information in occluded regions is usually tackled by imposing high regularization, resulting in oversmoothing the rest of the motion field. In case of very large displacements (acknowledged by s40+ metric), all the first methods (AggregFlow, [6, 53, 66, 82, 87, 89]) somehow integrate feature matching in their motion estimation process to capture the largest deformations. The high ranking of AggregFlow (ranked third) for this metric demonstrates the efficiency of the aggregation framework for integrating feature matching.

As for the *Final* set, AggregFlow is ranked fifth in terms of EPE-all. The slight decrease in performance compared to the Clean set is mainly due to errors caused by the added fog

[4] http://sintel.is.tue.mpg.de/

[5] http://vision.middlebury.edu/flow/

[6] http://www.cvlibs.net/datasets/kitti/

[7] http://lear.inrialpes.fr/src/epicflow/

[8] http://www.cse.cuhk.edu.hk/ leojia/projects/flow/

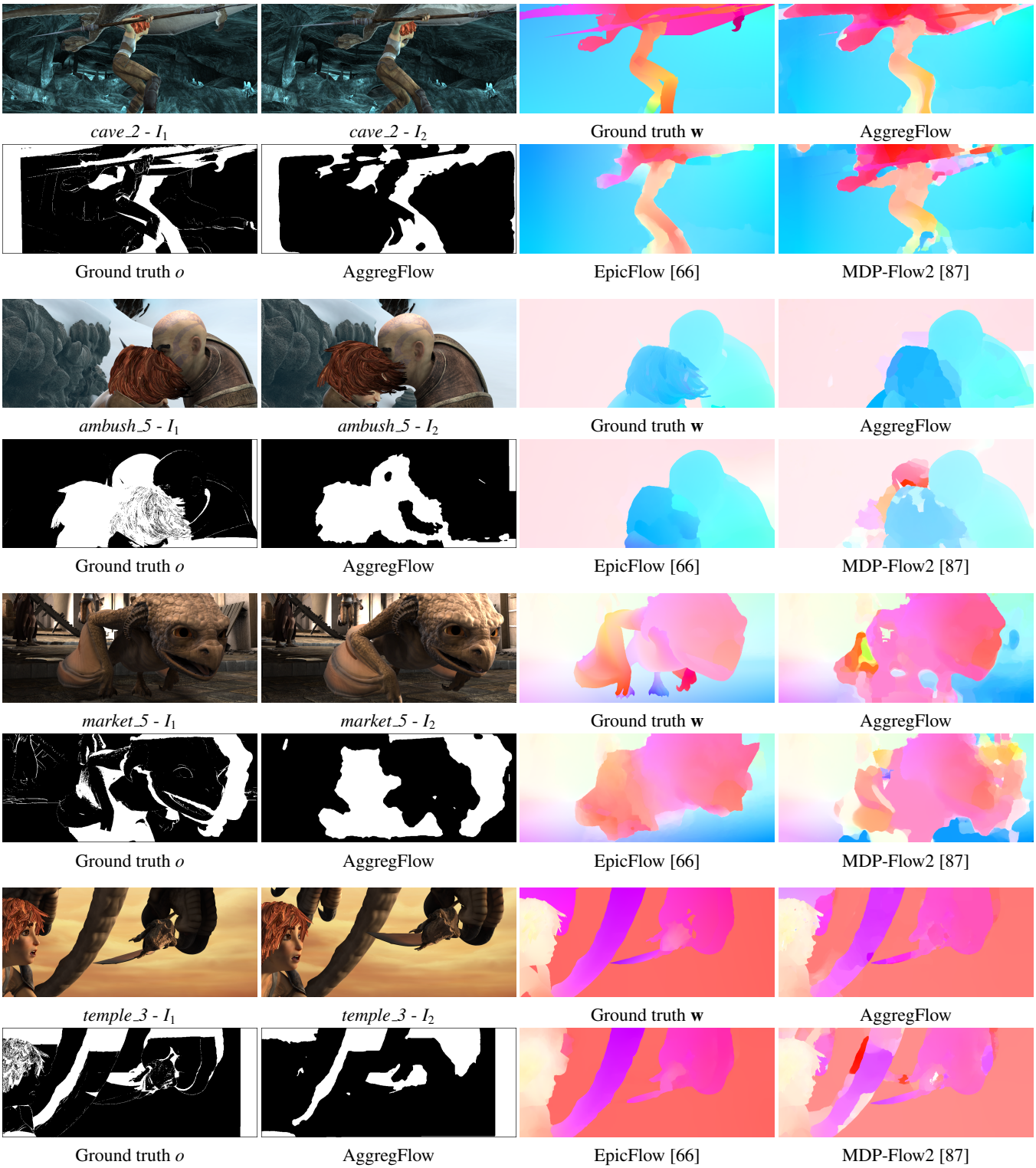


Figure 5: Comparative evaluation with EpicFlow [66] and MDP-Flow2 [87] on several sequences of the MPI Sintel dataset. Every first row from left to right: successive input images, ground truth motion field, motion field computed with AggregFlow. Every second row from left to right: ground truth occlusion, occlusion map computed with AggregFlow, motion fields computed respectively with EpicFlow [66] and MDP-Flow2 [87].

effect in the two *ambush* sequences. An even more pronounced performance decrease is also observed for PH-Flow [89]. As emphasized in [6], local intensity-based displacement computation tends to capture the motion of the fog rather than the movement of objects appearing in transparency. As our candidates estimation is local, it is subject to this limitation. Global variational approaches are able to diffuse motion estimates in these regions and are consequently better suited for

Table 3: Results on the MPI Sintel Clean test subset

	EPE all	EPE matched	EPE unmatched	d0-10	s40+
EpicFlow [66]	**4.115**	**1.360**	26.595	3.660	**25.859**
PH-Flow [89]	4.388	1.714	**26.202**	3.612	27.997
AggregFlow	4.754	1.694	29.685	3.705	31.184
TF+OFM [46]	4.917	1.874	29.735	3.676	31.391
SparseFlowFused [77]	5.257	1.627	34.834	4.211	33.489
DeepFlow [82]	5.377	1.771	34.751	4.519	33.701
PatchWMF-OF [78]	5.550	1.781	36.257	3.339	37.319
PCA-Layers [84]	5.730	2.455	32.468	5.447	35.079
LocalLayering [74]	5.820	2.143	35.784	3.817	39.976
MDP-Flow2 [87]	5.837	1.869	38.158	**3.210**	39.459
EPPM [6]	6.494	2.675	37.632	4.997	39.152
S2D-Matching [53]	6.510	2.792	36.785	5.523	44.187

Table 4: Results on the MPI Sintel Final test subset

	EPE all	EPE matched	EPE unmatched	d0-10	s40+
EpicFlow [66]	**6.285**	**3.060**	**32.564**	**5.205**	**38.021**
TF+OFM [46]	6.727	3.388	33.929	5.544	39.761
SparseFlowFused [77]	7.189	3.286	38.977	5.567	44.319
DeepFlow [82]	7.212	3.336	38.781	5.650	44.118
AggregFlow	7.329	3.696	36.929	5.538	44.858
PH-Flow [89]	7.423	3.795	36.960	5.550	44.926
S2D-Matching [53]	7.872	3.918	40.093	5.975	48.782
PCA-Layers [84]	7.886	4.256	37.480	7.284	47.449
PatchWMF-OF [78]	7.971	3.766	42.218	5.712	48.396
LocalLayering [74]	8.043	4.014	40.879	5.680	49.426
EPPM [6]	8.377	4.286	41.695	6.556	49.083
MDP-Flow2 [87]	8.445	4.150	43.430	5.703	50.507

this kind of situations. Despite this shortcoming, our method still yields significant improvement in unmatched regions and on motion discontinuities. One solution to improve results in fog regions would be to incorporate a more sophisticated feature correspondence technique as the ones proposed in [53, 82].

Middlebury dataset The Middlebury benchmark is composed of sequences with small displacements, where the main challenge is to be able to recover both complex smooth deformation, motion discontinuities and motion details. Table 5 reproduces results (if any) for the same methods as those listed for the MPI Sintel benchmark, since we consider that the latter is the prevailing benchmark, especially to evaluate methods on the currently most challenging issues, occlusion and large displacements. It can be observed that the average EPE-all values computed over the eight test sequences, together with the differences between methods, are much smaller than for the MPI Sintel dataset. The mean of the average EPE-all score computed over the compared methods in Table 4 is equal to 7.56 for the MPI Sintel Final subset and to 0.343 for the Middlebury dataset (from Table 5). We also provide the average rank over the 8 test sequences for each method which is the metric used for global ranking on the Middlebury website.

On the whole Middlebury benchmark, AggregFlow, at time of submission, is ranked 44 over 114 tested methods (which are not all published) in terms of average rank. The average rank is deduced from the ranks respectively obtained for the eight test sequences, each rank being established from the average endpoint error on the sequence. Let us emphasize that performances are very close in terms of average accuracy. For instance, the LSM method [44] ranked 25th, has an average EPE-all score of 0.316, which is only better than AggregFlow for 0.023. The difference between the EPE-all scores of AggregFlow and MDP-Flow2 [87] ranked second in the Middlebury benchmark is 0.094, whereas AggregFlow outperforms MDP-Flow2 with a difference of 1.093 in the MPI Sintel clean beanchmark. Let us mention that the top ranked published method is OFLAF [48] which has an average rank of 8.1 and an average EPE-all of 0.197 (OFLAF method was not tested on the MPI Sintel benchmark).

Table 5: Results (if available) on the Middlebury benchmark for the same set of methods as in Tables 3 and 4

	EPE all	Avg. rank
MDP-Flow2 [87]	**0.245**	**9.2**
PH-Flow [89]	0.265	22.6
EPPM [6]	0.329	38.7
AggregFlow	0.339	42.4
S2D-Matching [53]	0.347	40.9
EpicFlow [66]	0.392	53.0
TF+OFM [46]	0.417	56.8
DeepFlow [82]	0.416	58.8

A visual comparison with MDP-Flow2 [87] and EpicFlow [66] is provided in Fig.6. These sample results confirm the tightness of performance between methods on that dataset. Let us mention that the preservation of motion discontinuities with AggregFlow is more accurate than with the EpicFlow method and close to MDP-Flow2 performance. These results also show that AggregFlow is still competitive for recovering motion details in addition to the large velocities of the MPI Sintel benchmark.

KITTI dataset The sequences of the KITTI dataset [35] are recorded by a camera mounted on a moving vehicle. The displacements are only due to camera motion, which results in very specific diverging motion fields as illustrated in Figure 5.2. The best performing methods in this benchmark are dedicated to this particular motion type and consider additional information like multi-views or epipolar constraint.

A typical artifact generated by AggregFlow for this kind of sequences comes from the block artifacts usually generated by graph cut optimization (already identified in [60]), particularly prominent in case of smooth variations of the motion field, as it is the case in the KITTI benchmark.

We summarize in Table 6 results, when available, of the methods introduced for comparison on the MPI Sintel benchmark (Tables 3 and 4). We give the average end-point error over the whole image (EPE-all) and the percentage of erroneous pixels in non-occluded areas (Out-noc). The latter is the score used for the main ranking in the KITTI benchmark. Clearly, AggregFlow is less competitive on that particular benchmark. It is also the case for the EPPM method [6]. Several methods use sophisticated matching method or data constancy constraints to cope with the frequent intensity changes in the benchmark

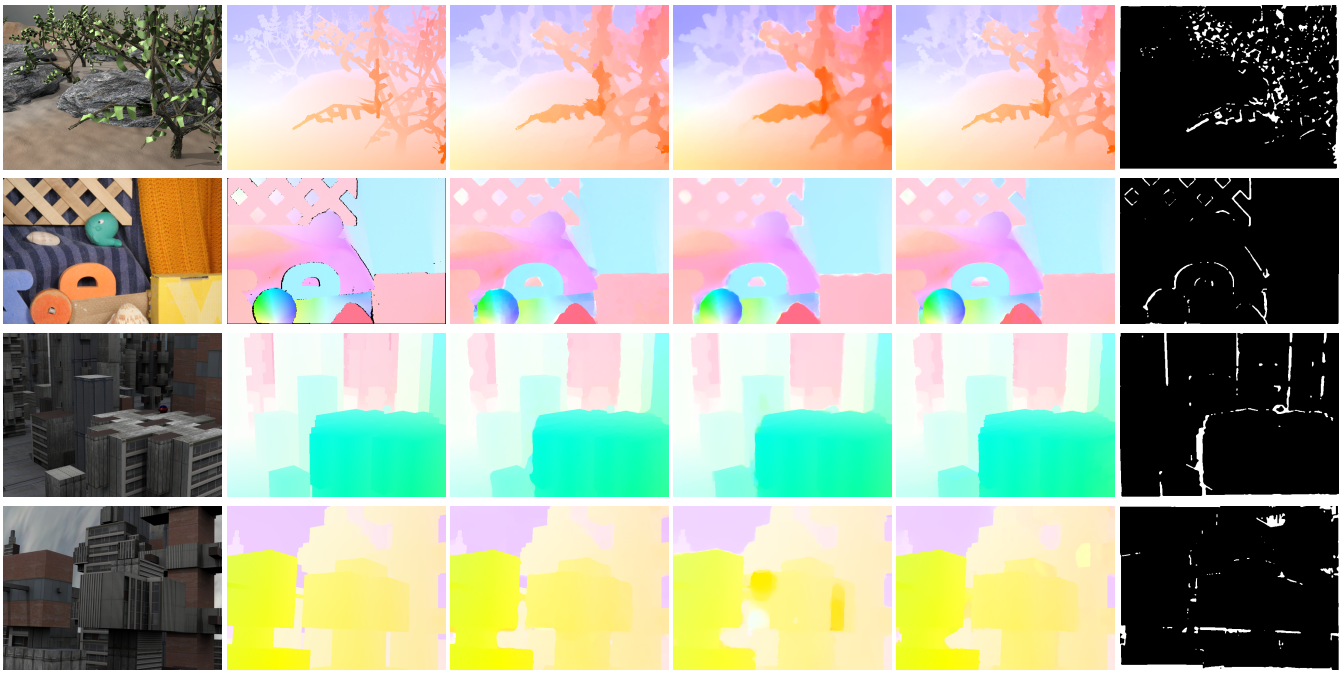


Figure 6: Comparative evaluation with [66] and [87] on several sequences of the Middlebury dataset. From top to bottom: sequences *grove3, rubberwhale, urban2, urban3*. In each row from left to right: first input image; ground truth motion field; motion field computed resp. with AggregFlow, EpicFlow [66] and MDP-Flow2 [87]; occlusion map computed with AggregFlow.

Table 6: Results (if available) on the KITTI benchmark for the same set of methods as in Tables 3 and 4

	EPE-all	Out-noc
PH-Flow [89]	2.9	5.76%
EpicFlow [66]	3.8	7.88%
TF+OM [46]	5.0	10.22%
PCA-Layers [84]	5.2	12.02%
DeepFlow [82]	5.8	7.22%
AggregFlow	7.4	12.23%
SparseFlow [77]	7.6	9.09%
EPPM [6]	9.2	12.75%

Image 1

Image 2

Motion field obtained with AggregFlow

Figure 7: Example of the estimated motion field for the KITTI benchmark.

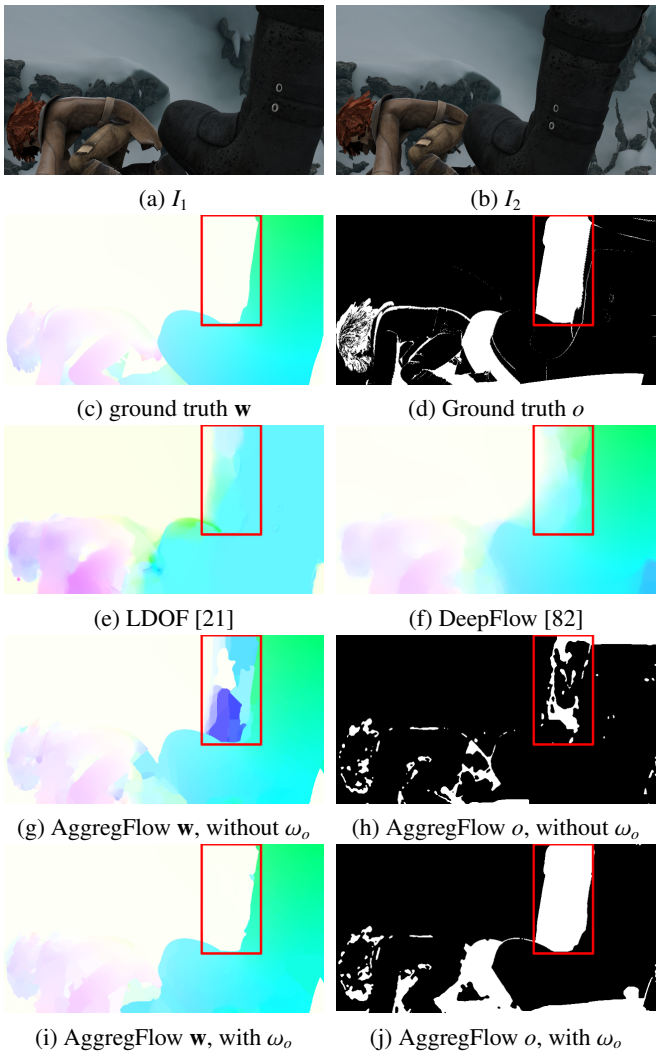
image sequences, and these techniques could be integrated in our aggregation framework to improve results. Another major problem on KITTI comes from the patch matching step, particularly affected by the scale change due to the zooming effect generated by the vehicle movement along the camera axis of view. Scale invariant patch matching should be implemented to cope with this problem. The best performing method on the KITTI benchmark among the methods which exploit only two frames and no epipolar constraint is PH-Flow [89]. The second best one is NLTGV-SC [65], with EPE-all score of 3.8px and Out-noc score of 5.93%). Performances of AggregFlow could be improved by adopting features of NLTGV-SC particularly well suited to the KITTI benchmark, like Total Generalized Variation reducing staircasing artifacts in smoothly varying motion fields, or scale invariant data conservation adapted to the zooming effect of the KITTI sequences.

5.3. Occlusion handling

The occlusion issue is nowadays one of the few main obstacles, if not the main, to improve optic flow methods. As aforementioned the impact of our occlusion framework on optical flow estimation was demonstrated by the EPE unmatched metric scores obtained on the MPI Sintel benchmark (Tables 3 and

Table 7: Results on MPI Sintel training sequences with progressive occlusion rates ("clean" pass), scores correspond to the EPE-all metric

Sequence name	*cave_4*	*ambush_5*	*market_6*	*cave_2*	*temple_3*	*ambush_6*	*ambush_2*
Occlusions rate	11%	14%	15%	16%	17%	18%	20%
AggregFlow	3.706	5.042	3.626	6.029	5.875	**5.854**	**5.632**
AggregFlow w/o occlusion	4.185	5.500	4.547	8.228	8.314	6.251	9.456
EpicFlow [66]	**3.597**	**4.541**	**3.138**	**5.707**	**4.520**	6.904	6.749
DeepFlow [82]	4.234	8.333	6.606	10.082	11.895	9.928	14.743
MDP-Flow2 [87]	3.815	6.591	5.384	8.347	9.011	8.466	12.083



(a) I_1 (b) I_2 (c) ground truth **w** (d) Ground truth o (e) LDOF [21] (f) DeepFlow [82] (g) AggregFlow **w**, without ω_o (h) AggregFlow o, without ω_o (i) AggregFlow **w**, with ω_o (j) AggregFlow o, with ω_o

Figure 8: Influence of the occlusion confidence map ω_o on motion and occlusion estimation. (e),(f): results of variational methods [21, 82] without occlusion handling. (g),(h): similar behaviour of our method without occlusion confidence map and impact on the occlusion detection. (i),(j): output of AggregFlow when integrating the occlusion confidence map.

4). Recovered occlusion maps displayed in Fig.5 and Fig.6 visually revealed the great ability of AggregFlow in coping with occluded regions. For the large occluded regions of Figure 5 for which ground truth is available, the estimated occlusion map is correct in most cases. A specific behaviour is noticeable in the *market_5* example, where occlusions are overdetected. It is due to the modeling assumption stating that occluded regions correspond to large violations of the data constancy equation. Regions of illumination changes may thus be detected as occlusions. While it leads strictly speaking to wrong occlusion detection, it can still be beneficial to motion estimation by implicitly treating illumination changes.

To complete the experimental evaluation of AggregFlow, we want now to further explore the performance of AggregFlow related to the occlusion issue. Since the occlusion framework is composed of several elements, we detail the influence of each one in the following. The efficiency of the motion candidates extension in occluded regions has already been highlighted in Section 3.3 and Table 1 through the analysis of the Best Candidate Flow.

We first investigate the role of the occlusion confidence map involved in the sparsity constraint (15). Illustrations are given in Fig.8. The results of two variational methods, LDOF [21] and DeepFlow [82], are also displayed in Fig.8 (e,f) for comparison. For these two methods, the motion subfield in the occluded region highlighted by the red bounding box, is wrongly estimated since no explicit occlusion detection is performed. If the occlusion map is initialized to $o(x) = 0$, $\forall x \in \Omega$, the occlusion terms of AggregFlow energy (11) are canceled in the very first iteration of the alternate optimization, which results in a similar behaviour as the one of LDOF and DeepFlow methods [21, 82]. If $\forall x \in \Omega, \omega_o(x) = 1$, the convergence remains trapped in the initial local minimum, as displayed in Fig.8 (g,h). The reason is that the occlusion map is strongly determined by the estimated motion field and cannot deviate from the output of the first iteration. The role of the confidence map ω_o is then to act as an additional evidence for occlusion detection, relaxing the coupling between **w** an o. The guidance of ω_o enables to deviate from the output of the first iteration and to converge to the result shown in Fig.8 (i,j).

We now focus on the evaluation of the occlusion model of the aggregation step. For this purpose, we distinguish between the full AggregFlow method, and AggregFlow without the occlusion-related terms in (11) removed by setting $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_4 = 0$. Nevertheless, the occlusion handling in AggregFlow first step is still kept for the production of motion candidates in occluded regions. To assess the impact of the occlusion rate on the method performance, we have selected training sequences of the MPI Sintel dataset for the "clean" pass, with progressive occlusion rates from 11% to 20%. Comparative evaluation between the two versions of AggregFlow and competitors of Table 3 with available code (EpicFlow [66], DeepFlow [82], and MDP-Flow2 [87]) is reported in Table 7. The improvement due to the occlusion terms in AggregFlow energy is clearly significant on all examples when comparing with Ag-

gregFlow without occlusion terms. AggregFlow performance is the second best one for the first occlusion rates while being close to EpicFlow (apart from the *temple-3* example), and is the best one for higher occlusion rates.

6. Conclusion

We have defined a two-step method for optical flow computation called AggregFlow which handles occlusion detection and occlusion filling with motion vectors in an original and efficient way. It yields accurate parametric motion candidates at every pixel in a first step, and resorts to a discrete optimization to aggregate sets of motion candidates into the global flow field. Our method can be viewed as a novel and efficient combination of local and global approaches for occlusion-aware optical flow computation. It articulates the computation of local motion candidates and their global aggregation while jointly recovering occlusion maps. The framework is generic, and both the local and global steps could be adapted for specific purposes.

We demonstrated the added value of combining patch correspondences and patch-based affine motion estimation to produce highly accurate motion candidates, advocating the relevance of patch-based parametric motion estimation, provided size and position of the patches are appropriately defined. The integration of multiple patch correspondences in the candidates generation process allows us to deal with local matching ambiguities. We formulated the aggregation step as a discrete optimization problem, selecting the best motion candidate at every pixel while preserving motion discontinuities and achieving occlusion recovery. The occlusion scheme acts in both steps of AggregFlow. An exemplar-based occlusion term is incorporated in the global aggregation energy. Incidentally, it could be integrated in other estimation paradigms as well, e.g., in variational approaches. Occlusion cues derived from the computed motion candidates are exploited in the sparse modeling of occlusions. Overall, AggregFlow achieves state-of-the-art results on the MPI Sintel benchmark. The most significant improvements are reached in occluded regions and for large displacements.

Extensions of the method could tackle remaining matching errors in the patch correspondence and in the exemplar search substeps. A more elaborate and discriminative distance than the pixel-based L_1 distance could be envisioned for patch matching. Future work could also deal with a GPU implementation to largely improve computation efficiency.

Acknowledgments

This work was realized as part of the Quaero program, funded by OSEO, French State agency for innovation. It was also partly supported by the France-BioImaging project granted by the "Investissement d'Avenir" national program.

References

[1] A. Alba, E. Arce-Santana, and M. Rivera. Optical flow estimation with prior models obtained from phase correlation. In *ISVC*, LNCS 6453, Las Vegas, November 2010.

[2] N. Apostoloff and A. Fitzgibbon, Learning spatiotemporal T-junctions for occlusion detection. In *CVPR*, San Diego, 2005.

[3] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. *Int. J. of Computer Vision*, 93(3):319-347, July 2011.

[4] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *Int. J. of Computer Vision*, 97:322--338, May 2012.

[5] S. Baker, D. Scharstein, JP Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *Int. J. of Computer Vision*, 92(1):1–31, 2011.

[6] L. Bao, Q. Yang and H. Jin. Fast edge-preserving PatchMatch for large displacement optical flow. In *CVPR*, Columbus, 2014.

[7] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image. In *SIGGRAPH*, New Orleans, August 2009.

[8] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43-77, 1994.

[9] B. Berkels, C. Kondermann, C. Garbe, and M. Rumpf. Reconstructing optical flow fields by motion inpainting. In *EMMCVPR*, Bonn, Aug. 2009.

[10] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester. Image inpainting. In *SIGGRAPH*, New Orleans, July 2000.

[11] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, Berlin, May 1993.

[12] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[13] M.J. Black and A.D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996.

[14] M.J. Black and D.J. Fleet Probabilistic detection and tracking of motion boundaries. *Int. J. of Computer Vision*, 38(3):231–245, 2000.

[15] A. Blake and A. Zisserman Visual reconstruction. Cambridge : MIT Press, 1987.

[16] M. Bleyer, C. Rhemann and M. Gelautz. Segmentation-based motion with occlusions using graph-cut optimization. In *DAGM*, Berlin, Sept. 2006.

[17] A. Bobick and S. Intille. Large occlusion stereo. *Int. J. of Computer Vision*, 33(3):181–200, 1999.

[18] Y. Boykov, O. Veksler, R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[19] J. Braux-Zin, R. Dupont and A. Bartoli. A general dense image matching framework combining direct and feature-based costs. In *ICCV*, Sidney, Dec. 2013.

[20] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, Prague, May 2004.

[21] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(3):500–513, March 2011.

[22] A. Bugeau, M. Bertalmío, V. Caselles and G. Sapiro. A comprehensive framework for image inpainting. *IEEE Trans. on Image Processing*, 19(10):2634-2645, Oct. 2010.

[23] D.J. Butler, J. Wulff, G.B. Stanley and M.J. Black. A natural-

istic open source movie for optical flow evaluation. In *ECCV*, Florence, Oct. 2012.

[24] T. Chan, S.H. Kang, and J.H. Shen. Euler's elastica and curvature based inpaintings. *SIAM Journal on Applied Mathematics*, 63(2), 564–592, 2002.

[25] T. Chan, S. Osher, and J.H. Shen. The digital TV filter and nonlinear denoising. *IEEE Trans. on Image Processing*, 10(2):231-241, 2001.

[26] L. Chen, H. Jin, Z. Lin, S. Coben, Y. Wu. Large displacement optical flow from nearest neighbor fields. In *CVPR*, Portland, June 2013.

[27] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *Int. J. of Computer Vision*, 62(3):246–265, 2005.

[28] A. Criminisi, P. Pérez and K. Tomaya. Region filling and object removal by exemplar-based image inpainting. In *IEEE Trans. on Image Processing*, 13(9):1200–1212, 2004.

[29] G. Facciolo, N. Limare and E. Meinhardt. Integral images for block matching. *IPOL*, 2013.

[30] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. of Computer Vision*, 70(1):41–54, 2006.

[31] D. Fortun and C. Kervrann. Semi-local variational optical flow estimation. In *ICIP*, Orlando, Sept. 2012.

[32] D. Fortun, P. Bouthemy, P. Paul-Gilloteaux, and C. Kervrann. Aggregation of patch-based estimations for illumination-invariant optical flow in live cell imaging. In *ISBI*, San-Francisco, April 2013.

[33] D. Fortun, P. Bouthemy, and C. Kervrann. Optic flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1-21, May 2015.

[34] W.T. Freeman and E.C. Pasztor, Learning low-level vision. In *ICCV*, Toronto, September 1999.

[35] A. Geiger, P. Lenz and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, Providence, June 2012.

[36] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33(4):725–740, April 2000.

[37] D. Hafner, O. Demetz, and J. Weickert. Why is the Census transform good for robust optic flow computation?. In *SSVM*, Leibnitz, Austria, LNCS 7893, Springer, June 2013.

[38] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Trans. on Patt. Anal. and Machine Intelligence*, 15(12):1217–1232, 1993.

[39] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[40] M. Hornáček, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using PatchMatch belief propagation. In *ECCV*, Zurich, LNCS 8691, Springer, Sept. 2014.

[41] S. Huq, A. Koschan and M. Abidi. Occlusion filling in stereo: Theory and experiments. *Computer Vision and Image Understanding*, 117(6):688–704, 2013.

[42] A. Humayun, O. Mac Aodha, and G.J. Brostow. Learning to find occlusion regions. In *CVPR*, Colorado Springs, 2011.

[43] S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *IEEE Trans. on Image Processing*, 17(8):1443–1451, 2008.

[44] K. Jia, X. Wang, and X. Tang. Optical flow estimation using learned sparse model. In *ICCV*, Barcelona, Nov. 2011.

[45] P.M. Jodoin and M. Mignotte. Optical-flow based on an edge-avoidance procedure. *Computer Vision and Image Understanding*, 113(4):511–531, 2009.

[46] R. Kennedy and C.J. Taylor Optical flow with geometric occlusion estimation and fusion of multiple frames. In *EMMCVPR*, Hong-Kong, LNCS 8932, Springer, January 2015.

[47] C. Kervrann, J. Boulanger, T. Pécot, P. Pérez, and J. Salamero. Multiscale neighborhood-wise decision fusion for redundancy detection in image pairs. *SIAM J. Multiscale Modeling & Simulation*, 9(4):1829–1865, 2011.

[48] T.H. Kim, H. Lee, and K.M. Lee. Optical flow via locally adaptive fusion of complementary data costs. In *ICCV*, Sydney, Dec. 2013.

[49] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[50] N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Trans. on Image Processing*, 16(11):2649–2661, Nov. 2007.

[51] C. Kondermann, D. Kondermann, and C. Garbe. Postprocessing of optical flows via surface measures and motion inpainting. In 30th DAGM Symposium on Pattern Recognition, LNCS 5096, Springer, Munich, June 2008.

[52] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, Anchorage, 2008.

[53] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *ICCV*, Sydney, December 2013.

[54] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, Vancouver, 1981.

[55] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Trans. on Image Processing*, 7(5):703–719, 1998.

[56] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: a synopsis of current problems and methods. *Int. J. of Computer Vision*, 19(1):29–55, July 1996.

[57] M. Mohamed, H. Rashwan, B. Mertsching, M. Garcia, and D. Puig. Illumination-robust optical flow using a local directional pattern. *IEEE Trans. on Circuits and Systems for Video Technology*, 24(9):1499–1508, Sept. 2014.

[58] M.G. Mozerov. Constrained optical flow estimation as a matching problem. *IEEE Trans. on Image Processing*, 22(5):2044–2055, 2013.

[59] C. Nieuwenhuis, D. Kondermann, C.S. Garbe. Complex motion models for simple optical flow estimation. DAGM Symp. on Pattern Recognition, LNCS 6376, Springer, Darmstadt, Sept. 2010.

[60] C. Nieuwenhuis, E. Toeppe and D. Cremers. A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model. *Int. J. of Computer Vision*, 104(3):223–240, Sept. 2013.

[61] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. of Visual Communication and Image Representation*, 6(4):348–365, 1995.

[62] J.M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.

[63] N. Papadakis, R. Yildizoglu, J.F. Aujol and V. Caselles. High-dimension multilabel problems: Convex or nonconvex relaxation? *SIAM J. on Imaging Sciences*, 6(4):2603–2639, 2013.

[64] F. Pierre, J.F. Aujol, A. Bugeau, N. Papadakis and V.T. Ta. Luminance-Chrominance Model for Image Colorization. *SIAM J. on Imaging Sciences*, 8(1):536–563, 2015.

[65] R. Ranftl, K. Bredies and T. Pock. Non-local total generalized variation for optical flow estimation. In *ECCV*, Zurich, September 2014.

[66] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, Boston, June 2015.

[67] D. Rosenbaum, D. Zoran, and Y. Weiss. Learning the local statistics of optical flow. In *NIPS*, Lake Tahoe, 2013.

[68] S. Roth and M.J. Black. Fields of experts. *Int. J. of Computer Vision*, 82(2):205-29, April 2009.

[69] C. Rother, V. Kolmogorov, V. Lempitsky and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, Minneapolis, June 2007.

[70] T. Senst, V. Eiselen and T. Sikora. Robust local optical flow for feature tracking. *IEEE Trans. on Circuits and Systems for Video Technology*, 22(9):1377–1387, Sept. 2012.

[71] A.N. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *Int. J. of Computer Vision*, 82:325–357, 2009.

[72] J. Sun, Y. Li, K. Sin., S,B, Kang and H-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, San Diego, June 2005.

[73] D. Sun, E. Sudderth and M. Black. Layered segmentation and optical flow estimation over time. In *CVPR*, Portland, June 2012.

[74] D. Sun, C. Liu, and H. Pfister. Local layering for joint motion estimation and occlusion detection. In *CVPR*, Columbus, June 2014.

[75] D. Sun, S. Roth, and M. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. of Computer Vision*, 106(2):115–137, Jan. 2014.

[76] Y. Tian and S.G. Narasimhan. Globally optimal estimation of nonrigid image distortion. *Int. J. of Computer Vision*, 98(3):278–302, 2012.

[77] R. Timofte and L. Van Gool. SparseFlow: Sparse matching for small to large displacement optical flow. In *WACV*, Lake Placid, Jan. 2015.

[78] Z.Tu, C. Van Gemeren, and R.C. Veltkamp. Improved color patch similarity measure based weighted median filter. In *ACCV*, Singapore, Nov. 2014.

[79] M. Unger, M. Werlberger, T. Pock and H. Bischof. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *CVPR*, Providence, June 2012.

[80] J. Ulén and C. Olsson. Simultaneous fusion moves for 3D-label stereo. In *EMMCVPR*, Springer, Lund, August 2013.

[81] A. Wedel, T. Pock, C. Zach, H. Bischof and D. Cremers An improved algorithm for TV-L1 optical flow. In *Dagstuhl Visual Motion Analysis Workshop*, 2008.

[82] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid. Large displacement optical flow with deep matching. In *ICCV*, Sydney, December 2013.

[83] M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *CVPR*, San Francisco, June 2010.

[84] J. Wulff and M.J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, Boston, June 2015.

[85] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, Graz, 2006.

[86] L. Xu, J. Chen and J. Jia. A segmentation-based variational model for accurate optical flow estimation. In *ECCV*, Marseille, October 2008.

[87] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(9):1744-1757, Sept. 2012.

[88] L. Xu, Z. Dai, and D. Jia. Scale invariant optical flow. in *ECCV*, Firenze, Sept. 2012.

[89] J. Yang and H. Li. Dense, accurate optical flow estimation with piecewise parametric model. In *CVPR*, Boston, June 2015.

[90] H. Zimmer, A. Bruhn and J. Weickert. Optic flow in harmony. *Int. J. of Computer Vision*, 93(3):368–388, 2011.

[91] C.W. Zitnick, N. Jojic and S.B. Sin. Consistent segmentation for optical flow estimation. In *ICCV*, Beijing, October 2005.

[92] Q. Zhang, L. Xu and J. Jia. 100+ times faster weighted median filter. In *CVPR*, Columbus, June 2014.